

WEST Search History

DATE: Wednesday, July 21, 2004

Hide?	<u>Set</u> <u>Name</u>	<u>Query</u>	<u>Hit</u> <u>Count</u>
	<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ</i>		
<input type="checkbox"/>	L37	l33 and l2	0
<input type="checkbox"/>	L36	l33 and l20	0
<input type="checkbox"/>	L35	l33 and l28	2
<input type="checkbox"/>	L34	byte adj2 packing adj2 technique	1
<input type="checkbox"/>	L33	byte adj2 packing	139
<input type="checkbox"/>	L32	L31 NOT l30	1
<input type="checkbox"/>	L31	L28 and (segmentation and reassembling)	3
<input type="checkbox"/>	L30	L28 and (segmenting) and (reassembling)	2
<input type="checkbox"/>	L29	L28 and (splitting) and (reassembling)	1
<input type="checkbox"/>	L28	20010817	2029
<input type="checkbox"/>	L27	((packet or stream)near2 (process or processing)).ti.	2841
<input type="checkbox"/>	L26	20010817	4154
<input type="checkbox"/>	L25	((packet or stream)near5 (process or processing)).ti.	5669
<input type="checkbox"/>	L24	L21 and l16	0
<input type="checkbox"/>	L23	L21 and l6	1
<input type="checkbox"/>	L22	L21 and l2	0
<input type="checkbox"/>	L21	20010817	42
<input type="checkbox"/>	L20	(process or processing) near8 (non-contiguous) near8 (data or stream or packet)	62
<input type="checkbox"/>	L19	20010817	1
<input type="checkbox"/>	L18	L16 and l6	4
<input type="checkbox"/>	L17	L16 and l1	0
<input type="checkbox"/>	L16	gap adj2 removing	816
<input type="checkbox"/>	L15	L14 and buffer	35
<input type="checkbox"/>	L14	L13 and l6	44
<input type="checkbox"/>	L13	20010817	909
<input type="checkbox"/>	L12	L11 and (stream adj3 (ID or identifier or identification or map or mapping or table))	1312
<input type="checkbox"/>	L11	stream near5 processing	22311
<input type="checkbox"/>	L10	20010817	10
<input type="checkbox"/>	L9	L8 and (data adj2 stream)	10

<input type="checkbox"/>	L8	(2:1 or 3:1 or 3:2) adj2 multiplexing	94
<input type="checkbox"/>	L7	L6 and l2	3
<input type="checkbox"/>	L6	(packet or uniot or cell)adj3 (split or splitter or splitting or separator or separating or strip or stripping or parse or parsing or remove or removing)	35784
<input type="checkbox"/>	L5	l3 NOT l4	4
<input type="checkbox"/>	L4	L2 and (stream adj3 (ID or iderntifier or identification))	2
<input type="checkbox"/>	L3	L2 and (packet adj3 (process or processing))	4
<input type="checkbox"/>	L2	20010817	41
<input type="checkbox"/>	L1	(data adj3 stream) near8 (multiplex or multiplexing) near8 bandwidth	66

END OF SEARCH HISTORY

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)[First Hit](#) [Fwd Refs](#)

Generate Collection

L14: Entry 42 of 44

File: USPT

Jul 15, 1997

DOCUMENT-IDENTIFIER: US 5648960 A

TITLE: Recording/reproducing apparatus for data packet stream

Application Filing Date (1):19951107Brief Summary Text (17):

Another object of the present invention is to provide a recording/reproducing apparatus for a data packet stream that is capable of stabilizing the processing of the decoder by controlling an output rate of the reproduced data.

Detailed Description Text (9):

A dummy packet producer 10 outputs a dummy packet that is in the same format as a packet in the input data packet stream and has identification data indicating that it is of a different type than the extracted data packets. For example, when an input data packet stream has MPEG2 transport data packets, these packets can be made hulled within the decoding process when the dummy packet producer 10 controls prescribed bits of the Link Level Header. For example, if a "1" is set in the error indicator, it is possible to make this packet null by considering it to be an error packet. Further, the PID may be set at a different value from the extracted data packets. In this case, the dummy packet is disregarded during the decoding process. Further, the dummy packet producer 10 may output a 188 byte padding packet having a PID that indicates that it is a dummy packet. The dummy packet producer 10 may set the payload to be null data (for example, "FF" in hexadecimal).

Detailed Description Text (10):

The dummy packet inserter 9 inserts a number of dummy packets, based on the number of the deleted packets in contiguous sequence, into the sequence of extracted data packets from the data separator 8. In short, when an extracted data packet is input, the dummy packet inserter 9 restores the extracted data packet to its original rate and location within the input data packet stream by inserting a dummy packet in accordance with the number of deleted packets that were next to the extracted data packets in the input data packet stream and outputs the data via output terminal 11 as an output data packet stream.

Detailed Description Text (11):

Next, the operation of the embodiment with the above-described construction will be explained with reference to FIGS. 2 and 3. FIG. 2(a) shows the input data packet stream. FIG. 2(b) shows the extracted data packets. FIG. 2(c) shows the number of deleted packets in contiguous sequence. FIG. 2(d) shows the output of the data combiner 4. FIG. 2(e) shows the output of the reproducing circuit 7. FIG. 2(f) shows the extracted data packets from the data separator 8. FIG. 2(g) shows the information relating of the number of deleted packets in contiguous sequence. FIG. 2(h) shows the output data packet stream. Further, the cross-hatching portions, as shown in FIG. 2, indicate dummy packets. FIG. 3 shows the output of the data combiner 4.

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)

THIS PAGE BLANK (USPTO)

[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)[First Hit](#)

Generate Collection

L14: Entry 3 of 44

File: PGPB

Sep 12, 2002

DOCUMENT-IDENTIFIER: US 20020128823 A1

TITLE: System and method for reception, processing and transmission of digital audio streamAbstract Paragraph:

A system and methods are described for processing digital audio stream data from received transport streams. A transport stream parser identifies particular transport packets related to audio stream data. The transport stream parser enables audio parser and provides packet identifiers of the particular transport packets to the audio parser. The audio parser selects the particular transport packets from a transport stream. The audio parser discards transport packets not related to specific audio types and provides packetized elementary stream audio data to an audio decoding system. The packetized elementary audio stream data is processed into an I2S elementary stream and decoded into pulse-coded modulation (PCM) audio data for output through an external audio receiver, such as through a digital to analog converter. The packetized elementary stream data and I2S elementary stream data are also stored in memory for playback requests generated by the audio decoding system at a later time.

Application Filing Date:

20010306

Summary of Invention Paragraph:

[0010] Prior art FIG. 3 illustrates the fields associated with a PES stream. Each PES stream contains a header portion and a data portion. In addition, an optional header portion may exist. The header portion includes a Packet Start Prefix, a stream ID, and a packet length indicator.

Detail Description Paragraph:

[0036] A transport stream adaptation field parser (TSAFP) 530 may be used to select data from transport packets identified as having specific adaptation field data as shown in Prior-art FIG. 1. In one embodiment, TSAFP 530 is used to extract program clock reference (PCR) values, found within the adaptation field. The PCR values may be used to synchronize a local system clock, such as system time clock (STC) counter 546. In one embodiment, the extracted PCR values are stored in a PCR register, such as AF_PCR_REG register 532. The PID values related to transport packets used to extract PCR values from may be stored in a separate register, such as PCR_PID_REG register 514. In one embodiment, TSAFP 520 is enabled through TPP 520. In other embodiments, TSAFP compare the transport packet PIDS to a predefined PID to determine which packets to parse. Specific states of operation of TSAFP 520 are provided in more detail in reference to FIG. 10.

Detail Description Paragraph:

[0047] In one embodiment, transport packets with PID values matching AUDIO_PID 693 are passed to PES extractor 640. In one embodiment, audio data is found within a payload section of the transport packet passed by audio PID filter 630. PES extractor 640 passes the audio data to audio PES parser 650. Audio PES parser 650 performs analysis on the data related to the received transport packet to determine whether or not to pass the data. In one embodiment, audio PES parser 650 includes

*parsing for audio packets
with headers including info
alt. stream*

an audio stream ID filter 660. Audio stream ID filter 660 compares a stream ID field within the PES audio data (as shown in Prior-art FIG. 3) to a list of known stream ID values. In one embodiment, the known stream ID values indicate particular audio stream types that are supported by the processing system, or an external audio decoder. For example, in one embodiment, only PES packets with stream ID values within the range of 0xC0-0xDF, or indicating a specific private stream ID, may be processed and packets with stream ID values outside of those ranges are discarded by audio stream ID filter 630.

Detail Description Paragraph:

[0048] An audio stream filter selector 662 may be used to enable or disable audio stream ID filter 660. An AUDIO_STREAM_PROCESS_ID register 694 may be monitored to provide an indication to audio stream filter selector 662, whether or not to allow the PES data to be processed through audio stream ID filter 660. In one embodiment, data that has been passed may be sent to an external I2S audio decoder. A MASK_I2S_REQ component 670 may be used to enable or disable the output of the PES data to the external I2S decoder. In one embodiment, a MASK_I2S_REQ register value is set to enable or disable the output.

Detail Description Paragraph:

[0054] A TD_PES register may be used to provide information related to parsed PES data. A PES_STREAM_ID field may be used to provide a stream ID value extracted from a PES packet currently being parsed. A PES_HEADER_DATA_LEN field may be used to provide the length of a PES header related to a PES packet being parsed. A TD_PES_EXT register may be used to provide information related to extension data provided with PES data being parsed. For example, a PES_EXT_FIELD_LEN field indicates the length of the PES packet extension data bit-field.

Detail Description Paragraph:

[0069] In one embodiment, an AUDIO_BUFFER_INDEX field may be used to identify one of sixty-four buffers in system memory where audio data may be routed. An AUDIO_START_FROM_PUSI field may be used to indicate whether the audio parser starts from a current transport packet or from a transport packet which includes a payload unit start indicator set to a value of '1'. An AUDIO_PROCESS_STREAM_ID field may be used for disabling filtering to be performed based on a stream ID indicator. An AUDIO_STREAM_ID field may indicate a particular stream ID value to filter on. While particular transport packets may be flagged due to errors associated with the transport packet during processing, an IGNORE_AUDIO_TEI may be set to configure the audio parser to ignore all error flagged transport streams.

CLAIMS:

20. The method as in claim 19, wherein said audio decoding system includes an elementary stream formatter for processing data associated with the data word into an elementary stream.

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)

[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)[First Hit](#)

Generate Collection

L4: Entry 2 of 2

File: DWPI

Mar 30, 2001

DERWENT-ACC-NO: 2002-148502

DERWENT-WEEK: 200220

COPYRIGHT 2004 DERWENT INFORMATION LTD

TITLE: Multiplexing higher bandwidth data stream and several lower bandwidth data streams by embedding clock reference data in each stream header as timing reference for reconstitution of lower bandwidth data streams at receiver

Basic Abstract Text (3):

ADVANTAGE - Allows a single physical link to transmit, route and switch a complete family of existing 8-bit and 10-bit signal formats and various generic data formats. Existing equipment for serializing and de-serializing high-definition bit-parallel data signals for distribution throughout a facility can be used to serialize and de-serialize the high-bandwidth multiplexed data signal without any additional hardware other than equipment for multiplexing and de-multiplexing between bit-parallel high- and low-bandwidth formats word providing a data valid bit or 'status bit' for each of the channels, which identifies valid data words within the packet. The source data streams are accurately reconstituted by de-multiplexing according to header data specifying routing destination, stream ID, stream type, channel usage and stream clock re-clock reference for clock recovery at the receiving end.

PF Application Date (1):19990930Standard Title Terms (1):

MULTIPLEX HIGH BANDWIDTH DATA STREAM LOWER BANDWIDTH DATA STREAM EMBED CLOCK REFERENCE DATA STREAM HEADER TIME REFERENCE RECONSTITUTED LOWER BANDWIDTH DATA STREAM RECEIVE

[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

[Previous Doc](#)[Next Doc](#)
[First Hit](#)[Go to Doc#](#)

Generate Collection

L7: Entry 1 of 3

File: PGPB

Oct 31, 2002

DOCUMENT-IDENTIFIER: US 20020162114 A1

TITLE: System and method for multicasting packets in a subscriber network

Application Filing Date:
20010430Summary of Invention Paragraph:

[0005] Content and data providers provide streams of data, data streams, that include video, audio and data, to cable operators via video sources, such as video encoders and video servers. The data streams are initially prepared for transmission through the broadband system by programming, or mapping, the video, audio and data with control software within a digital network control system (DNCS), which is an element manager for processing data within the headend. The DNCS causes the data streams associated with several programs to be combined into bundled groups of sessions. More specifically, the cable operator defines and maps the specifications of the individual data streams from one or several content and data providers and, for example, multiplexes them into grouped sessions in order to maximize the use of the bandwidth available within the cable television system.

Detail Description Paragraph:

[0139] The packet requester 740 reads the modulator identifier 804 of the DUH 802 to determine which modulator(s) 744 is to receive the data unit packet 800. The packet requester 740 also removes the DUH 802 from data unit packet 800, thereby reverting the data unit packet 800 into a standardized transport stream packet 500 for transmission. If the packet 800 is a unicast packet the packet requester 740 puts the transport stream packet 500 in the output buffer indicated by the modulator identifier 804. However, if the modulator identifier 804 indicates more than one output buffer, then the data unit packet 800 is a multicast packet, and the packet requester 740 sends a copy of the transport stream packet 500 to each output buffer indicated in the modulator identifier 804.

[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)[First Hit](#)

Generate Collection

L14: Entry 11 of 44

File: PGPB

Jan 31, 2002

DOCUMENT-IDENTIFIER: US 20020012528 A1

TITLE: Recording method of stream data and data structure thereof

Application Filing Date:

20010315

Brief Description of Drawings Paragraph:

[0052] FIG. 15 is a view for explaining the internal data structure of a stream file information table (SFIT);

Detail Description Paragraph:

[0111] More specifically, as shown in FIG. 3(f), STREAM.IFO (or SR_MANGR.IFO) 105 is comprised of video manager (VMGI or STR_VMGI) 231, stream file information table (SFIT) 232, original PGC information (ORG_PGCI) 233, user-defined PGC information table (UD_PGCIT) 234, text data manager (TXTDT_MG) 235, and manufacturer information table (MNFIT) or application private data manager (APDT_MG) 236 that manages navigation data SR PRIVT.DAT 105a unique to an application.

Detail Description Paragraph:

[0112] Stream file information table (SFIT) 232 shown in FIG. 3(f) can contain stream file information table information (SFITI) 241, one or more pieces of stream object information (SOBI) #A.cndot.242, #B.cndot.243, . . . , original PGC information general information 271, and one or more pieces of original cell information #1.cndot.272, #2.cndot.273, . . . , as shown in FIG. 3(g).

Detail Description Paragraph:

[0123] Two pieces of management information (STREAM.IFO 105) for stream objects #A.cndot.298 and #B.cndot.299 in FIG. 4 are recorded as two pieces of stream object information (SOBI) #A.cndot.242 and #B243 in stream file information table (SFIT) 232, as shown in FIG. 3(f) and (g).

Detail Description Paragraph:

[0235] The PES header includes packet start code prefix information, stream ID (private stream 2) information, and PES packet length information, as shown in FIG. 8(f).

Detail Description Paragraph:

[0236] The substream ID has contents for specifying stream recording data, as shown in FIG. 8(f). More specifically, substream ID="00000010b" indicates that data stored in that stream pack is stream recording data. When this stream ID is "10111110b", it indicates that the stream pack of interest is used as a padding packet (see FIG. 6(g)).

Detail Description Paragraph:

[0254] On the other hand, as for the subsequent stream packet in the SOB, an application packet may be segmented (split) at the boundary of neighboring stream packets.

Detail Description Paragraph:

[0255] Two transport packets 1k shown in FIG. 1(g) or partial application packets shown in FIG. 5(h) and (i) indicate application packets formed by this segmentation (split).

Detail Description Paragraph:

[0266] Optical disc device 415 comprises recording/playback unit 409 including a disc drive, data processor (to be abbreviated as D-PRO hereinafter) 410 for processing stream data to recording/playback unit 409 (or stream data from recording/playback unit 409), temporary storage 411 for temporarily storing stream data that overflows from D-PRO 410, and optical disc device controller 412 for controlling operations of recording/playback unit 409 and D-PRO 410.

Detail Description Paragraph:

[0308] As shown in FIG. 3(i), time map information 252 records only time stamp differential time information of each stream block. Hence, the values of time differences of stream blocks in time map information 252 are summed up in each of stream object information 242 or 243, and comparison must be made to check if this summed-up value has reached the time stamp time designated by STB unit 416. Based on the comparison result, the position of a stream block in a stream object, which block includes the time stamp value that matches the time designated by STB unit 416, is detected.

Detail Description Paragraph:

[0339] This streamer information STRI is comprised of streamer video manager information STR_VMGI, stream file information table SFIT, original PGC information ORG_PGCi (more generally, PGC information PGCi#i), user-defined PGC information table UD_PGCIT, text data manager TXTDT_MG, and application private data manager APDT_MG, as shown in FIG. 3 (f) or FIG. 13.

Detail Description Paragraph:

[0342] Stream file information table SFIT includes all navigation data that directly pertain to the streamer operation. Details of stream file information table SFIT will be explained later with reference to FIG. 15.

Detail Description Paragraph:

[0390] FIG. 15 is a view for explaining the internal data structure of the stream file information table (SFIT).

Detail Description Paragraph:

[0391] As shown in FIG. 15, stream file information table SFIT is made up of stream file information table information SFITI, one or more pieces of stream object stream information SOB_STI#n, and stream file information SFI.

Detail Description Paragraph:

[0392] Stream file information table information SFITI consists of the number SFI_Ns of pieces of stream file information on information storage medium (DVD-RAM disc) 201, the number SOB_STI_Ns of pieces of stream object stream information that follow SFITI, end address SFIT_EA of SFIT, and start address SFI_SA of SFI.

Detail Description Paragraph:

[0435] Streamer information STRI included in management information 105 contains stream file information table SFIT that manages stream object SOB which forms the contents of stream data. This SFIT includes stream object information SOBI that manages SOB. This SOBI includes access unit general information AU_GI including management information (access unit start map AUSM), and management information (PTSL).

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)

[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)[Search Forms](#)[Search Results](#)[Help](#)[Generate Collection](#)[User Searches](#)[Entry 14 of 44](#)

File: PGPB

Aug 16, 2001

[Preferences](#)[Logout](#)

DOCUMENT-IDENTIFIER: US 20010014058 A1

TITLE: Stream data generation method and partial erase processing method

Application Filing Date:20010315Brief Description of Drawings Paragraph:

[0076] FIG. 29 is a view for explaining the internal data structure of a stream file information table (SFIT in FIG. 3 or FIG. 27);

Detail Description Paragraph:

[0081] A stream data generation method, its recording method, a partial erase processing method of recorded stream data, and so on according to embodiments of the present invention will be described hereinafter with reference to the accompanying drawings.

Detail Description Paragraph:

[0116] More specifically, as shown in FIG. 3(f), STREAM.IFO (or SR_MANGR.IFO) 105 is comprised of video manager (VMGI or STR_VMGI) 231, stream file information table (SFIT) 232, original PGC information (ORG_PGCI) 233, user-defined PGC information table (UD_PGCIT) 234, text data manager (TXTDT_MG) 235, and manufacturer information table (MNFIT) or application private data manager (APDT_MG) 236 that manages navigation data SR_PRIVT.DAT 105a unique to an application.

Detail Description Paragraph:

[0117] Stream file information table (SFIT) 232 shown in FIG. 3(f) can contain stream file information table information (SFITI) 241, one or more pieces of stream object information (SOBI) #A.cndot.242, #B.cndot.243, . . . , original PGC information general information 271, and one or more pieces of original cell information #1.cndot.272, #2.cndot.273, . . . , as shown in FIG. 3(g).

Detail Description Paragraph:

[0125] Two pieces of management information (STREAM.IFO 105) for stream objects #A.cndot.298 and #B.cndot.299 in FIG. 4 are recorded as two pieces of stream object information (SOBI) #A.cndot.242 and #B.cndot.243 in stream file information table (SFIT) 232, as shown in FIGS. 3(f) and (g).

Detail Description Paragraph:

[0194] Separator 425 transfers video packet data (MPEG video data) to video decoder 428, audio packet data to audio decoder 430, and sub-picture packet data to SP decoder 429 in accordance with stream ID 603 and substream ID to be described later with reference to FIG. 10.

Detail Description Paragraph:

[0236] PES header 601 in FIG. 10(a) includes packet start code prefix 602, stream ID 603, playback time stamp 604, and the like, as shown in FIG. 10(b). This PES header 601 corresponds to PES headers 6 to 9 shown in FIGS. 1(c), (i), and (j), PES headers 6 and 7 in FIG. 8(h), PES header 6 shown in FIG. 9, and the like.

Detail Description Paragraph:

[0237] A stream PES header in FIG. 10(d) includes a packet start code prefix, stream ID (private stream 2), PES packet length, stream ID, and the like. This stream PES header is the same as that shown in FIG. 9, and has contents corresponding to PES header 601 in FIG. 10(a).

Detail Description Paragraph:

[0238] When PES header 9 in FIG. 1(j) has the internal structure of PES header 601 shown in FIG. 10(a), if stream ID 603 (FIG. 10(b)) of this PES header is "10111110", a packet having this PES header is defined to be padding packet 40 (FIG. 1(i)) in MPEG.

Detail Description Paragraph:

[0282] In this case (YES in step S06), the value of stream ID 603 in FIG. 10(b) is set to be "10111110", as described above, and sector No. 79 (a sector entirely stuffed with a padding area) is converted into padding packet 40 (step S07).

Detail Description Paragraph:

[0292] The stream data playback sequence will be explained below while focusing on a process for extracting transport packets in separator 425 in FIG. 7 from stream information recorded on information storage medium (DVD-RAM disc) 201 with the structure shown in FIGS. 1(c) and (i) or FIG. 8(h).

Detail Description Paragraph:

[0295] Main MPU 404 in FIG. 7 then accesses stream file information table (SFIT) 232 in FIG. 3(f) in accordance with control program "stream data playback controller 404c" to read the contents of time map information 252 in FIG. 3(h). Main MPU 404 detects the number of a stream block (SOBU) that includes the position (playback start time position) of the designated "playback start time" and the start position address of that stream block (SOBU) (step S11).

Detail Description Paragraph:

[0308] The packet data in the converted data sequence include video packets, sub-picture (SP) packets, audio packets, and the like in correspondence with the contents upon recording. Each data pack including such packets has a pack header, and the type of data (video, sub-picture, audio, or the like) can be discriminated by the stream ID (not shown) in that pack header.

Detail Description Paragraph:

[0309] With reference to the contents of the stream ID, each video packet is transferred to video decoder 428 in FIG. 7, each sub-picture packet to SP decoder 429, and each audio packet to audio decoder 430. In this way, the respective decoders (428 to 430) individually decode the corresponding recorded contents (step S19).

Detail Description Paragraph:

[0448] On the other hand, as for the subsequent stream packet in the SOB, an application packet may be segmented (split) at the boundary of neighboring stream packets. The partial packet exemplified in the lower portion of FIG. 9 indicates an application packet formed by this segmentation (split).

Detail Description Paragraph:

[0472] On the other hand, as for the subsequent stream packet in the SOB, an application packet may be segmented (split) at the boundary of neighboring stream packets. The partial application packet shown in FIGS. 8(f) and (g) or FIG. 9 indicates an application packet formed by this segmentation (split).

Detail Description Paragraph:

[0499] This streamer information STRI is comprised of streamer video manager information STR_VMGI, stream file information table SFIT, original PGC information

ORG_PGCI (more generally, PGC information PGCi#i), user-defined PGC information table UD_PGCIT, text data manager TXTDT_MG, and application private data manager APDT_MG, as shown in FIG. 3(f) or FIG. 27.

Detail Description Paragraph:

[0502] Stream file information table SFIT includes all navigation data that directly pertain to the streamer operation. Details of stream file information table SFIT will be explained later with reference to FIG. 29.

Detail Description Paragraph:

[0552] FIG. 29 is a view for explaining the internal data structure of the stream file information table (SFIT in FIG. 3(f) or FIG. 27).

Detail Description Paragraph:

[0553] As shown in FIG. 29, stream file information table SFIT is made up of stream file information table information SFITI, one or more pieces of stream object stream information SOB_STI#n, and stream file information SFI.

Detail Description Paragraph:

[0554] Stream file information table information SFITI consists of the number SFI_Ns of pieces of stream file information on information storage medium (DVD-RAM disc) 201, the number SOB_STI_Ns of pieces of stream object stream information that follow SFITI, end address SFIT_EA of SFIT, and start address SFI_SA of SFI.

Detail Description Paragraph:

[0611] When temporary erase is executed for a middle portion (a range from APAT=A to APAT=B) of this program #j, originally recorded cell #k is split into three cells, i.e., cell #k (TE flag="0b"; start time=SC_S_APATk; end time=SC_E_APATk), cell #k+1 (TE flag="10b"; start time=SC_S_APATk+1; end time=SC_E_APATk+1), and cell #k+2 (TE flag="00b"; start time=SC_S_APATk+2; end time=SC_E_APATk+2).

Detail Description Paragraph:

[0653] When stream ID 603 (or substream ID) of PES header 601 in FIG. 10 (or stream PES packet header) is set at "10111110" to define sector No. 79 as padding packet 40, even when it is erroneously interpreted that stream data is recorded in padding area 38 and data transfer is made, the encoder unit (video encoder 416, audio encoder 417, and SP encoder 418) can understand and skip padding packet 40.

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)
[First Hit](#) [Fwd Refs](#)

Search Forms

Search Results

Help



Generate Collection

User Searches

Preferences Entry 21 of 44

File: USPT

Jul 13, 2004

Logout

DOCUMENT-IDENTIFIER: US 6763390 B1

TITLE: Method and system for receiving and framing packetized data

Application Filing Date (1):
20000124

Brief Summary Text (12):

Prior art FIG. 3 illustrates the fields associated with a PES stream. Each PES stream contains a header portion and a data portion. In addition, an optional header portion may exist. The header portion includes a Packet Start Prefix, a stream ID, and a packet length indicator.

Detailed Description Text (12):

In response to being enabled by the TPP, the Video PES Parser 430 further processes the packet by parsing the header of the video PES. Based upon information carried in the header of the video PES, registers are updated, and the video payload may be stored or discarded.

Detailed Description Text (97):

In response to receiving the first byte of the packet, the TPP will parse the packet header at step 213. From the parsing of the header, the TPP will retrieve the PID value of the packet.

Detailed Description Text (100):

When the PID allocation table, or other means, indicates the packet is a video packet the Packetized Elementary Stream Parser (PESP) is enabled to allow further processing. In the specific embodiment of the PID allocation table listed above, the video PID is stored as PID_0. However, other methods of identifying the video PID, such as the use of a flag or other indicator are also possible. The operation of the PESP is controlled by the PESP Control Registers, as illustrated in FIG. 18.

Detailed Description Text (102):

In the implementation of FIG. 21, a storage location within the storage locations 751 is reserved for the STREAM ID header field of a transport stream packet. In the embodiment shown at FIG. 21, inputs of the storage location for STREAM ID are connected to the appropriate bits of the data bus and the counter controller 752, to receive stream ID data from the FRAMER DATA representation of the transport stream at the correct time. The counter controller 752 receives the VSTART signal indicating the start of a new video PES and generates enable signals to capture the stream ID, and other information, from the video PES header. The counter controller is controlled by the signals VIDEO, FRAMER DEN, and VSTART. The VIDEO signal indicates the current packet is a video packet. The FRAMER DEN signal indicates when the current FRAME DATA byte is valid, and VSTART indicates when the current packet is the first packet for the video PES, in other words, VSTART indicates when the video packet contains video PES header data to be parsed. Based upon the VSTART signal, and the FRAMER DEN signal, the counter controller 752 can determine which byte of the header is currently being received.

Detailed Description Text (105):

The video data control module 755 contains logic that enables the video data payload of the present packet to be stored. Operation of the control module 755 is determined in the content of various video control registers, as illustrated in FIG. 18. The EnableParsing field is a one bit field, which when negated prevents any data from the current video packet from being saved. The ProcessStreamID field is 1 bit-field. When asserted, it enables further parsing based upon a specific stream ID value of the video PES header, such that if the video control register field StreamID of FIG. 18, does not match the parsed steam ID from the current packet, the data of the packet will not be saved. This is an advantage over the prior art, where filtering on the stream ID field of the video PES was done in software, generally by the system.

Detailed Description Text (112):

At step 219, a determination is made whether the current video packet is to be parsed based upon the packet stream ID. If so, flow proceeds to step 220, if not, flow proceeds directly to step 223.

Detailed Description Text (113):

At step 220, the PESP parses the stream ID from the PES header as discussed with reference to FIG. 21. In addition, FIG. 23 illustrates addition hardware parsing which can be performed by the PESP.

Detailed Description Text (119):

The counter controller 752 will generate enable signals to capture the various PES header fields based upon the values stored in locations 736, and a counter value generated by counter 737. The counter controller is controlled by the signals VIDEO, FRAMER DEN, and VSTART. The VIDEO signal indicates the current packet is a video packet. The FRAMER DEN signal indicates when the current FRAME DATA byte is valid, and VSTART indicates when the current packet is the first packet for the video PES, in other words, VSTART indicates when the video packet contains video PES header data to be parsed. Based upon the VSTART signal, and the FRAMER DEN signal, the counter controller 752 can determine which byte of the header is currently being received. As indicated with reference to the StartFromPUSICCommand register of FIG. 18, the counter controller can either allow for immediate PES parsing upon receiving a video packet, or it can wait to parse the PES data until a packet containing PES header information is received. Where PES parsing is immediate, the video data is provided to the output buffer.

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)
[First Hit](#) [Fwd Refs](#)



Generate Collection

L14: Entry 43 of 44

File: USPT

Sep 28, 1993

DOCUMENT-IDENTIFIER: US 5249292 A

TITLE: Data packet switch using a primary processing unit to designate one of a plurality of data stream control circuits to selectively handle the header processing of incoming packets in one data packet stream

Abstract Text (1):

A high speed data packet switching circuit has a software controlled primary processing unit, a plurality of network interface units connected to a plurality of networks for receiving incoming data packet streams and for transmitting outgoing data packet streams, a plurality of high speed data stream hardware control circuits for processing data packets in response to instructions from the primary processing unit and circuitry for interconnecting the primary processing unit, the interface units, and the data stream control circuits. The primary processing unit receives from the network interface unit at least a first one of the data packets of each new data packet stream and assigns that stream to be processed by one of the data stream control circuits without further processing by the primary processing unit. The apparatus and method thus perform routine, repetitive processing steps on the further packets of the data stream using the high speed hardware circuitry, while the initial processing and other non-repetitive or special processing of the data packets are performed in software. Particular hardware is described for effecting the high speed hardware processing of the data packets.

Application Filing Date (1):
19920310

Brief Summary Text (14):

The use of multiple processors to avoid the "processor bottleneck" provides some gains but again has limits. Given a code path to forward a packet, it is not plausible to split that path into more than a few stages. Three is typical: network input; protocol functions; and network output. The basis for this limitation is the overhead incurred to interface the different processors beyond a limited number of task divisions; that is, after a certain point, the increase in interface overhead outweighs the savings obtained from the additional stage. This is particularly true because of the need to tightly integrate the various components, for example, congestion control at the protocol level requires close coordination with the output device. Also, the interface overhead costs are made more severe by the complication of the interface which is required.

Brief Summary Text (19):

The invention relates to a method and apparatus for effecting high speed data packet switching. The switching circuit features a software controlled primary processing unit; a plurality of network interface units for receiving incoming data packet streams and for transmitting outgoing data packet streams from and to network paths respectively; a plurality of data stream control circuits or flow blocks for processing data packets in response to the primary processing unit; and circuitry for interconnecting the primary processing unit and the plurality of interface units and data stream control circuits. The primary processing unit is adapted to receive from the network interface units, and to process, at least a

first one of the data packets of each new data packet stream and to assign this stream to be processed by a data stream control circuit without further intervention or processing by the primary processing unit. It is important to note that this first packet is not necessarily a "connection set up" packet or any other similar explicit direction to the switch to set up a stream. Rather, as is usual in the connectionless datagram model, this first packet is just another user data packet.

Brief Summary Text (20):

In particular aspects of the invention, the data stream control circuit features a pattern matching circuit, responsive to pattern setting signals from the primary processing unit and to the incoming data packets from the network interface units, for identifying those packets of a packet stream which will be processed by the control circuit. The data stream control circuit further features a processing unit responsive control circuit for controlling, in response to control signals sent by the primary processing unit, the congestion control and header modification, stripping and prepending functions of the data stream control circuit. The data stream control circuit further features a data buffer responsive to the pattern matching circuitry and the processing unit responsive control circuit for storing data and protocol elements of an incoming data packet stream and for outputting a data packet stream to be forwarded along a communications path.

Brief Summary Text (21):

The network interface unit features, in one aspect of the invention, a network interface circuit for communicating with a network channel and an interface adapter for receiving channel data from the network interface circuit and for transmitting that channel data over the interconnecting circuit structure to the data stream control circuits and the primary processing unit, and for receiving network data from the data stream control circuits and the primary processing unit over the interconnecting circuit structure and for providing received data to the associated network interface circuit for transmission over a network channel.

Brief Summary Text (23):

The method of the invention features the step of separating from a software controlled primary processing unit used in a high speed data packet switching circuit a portion of the functionality which is repetitively used in connection with the processing of the second and further packets of an input data stream and implementing that portion of the functionality in hardware elements.

Detailed Description Text (10):

Referring to FIG. 1, in accordance with a particular embodiment of the invention, a specialized hardware 10 does all the work necessary for forwarding a "normal" packet in a previously identified packet stream from one network interface to another. All packets which the specialized hardware 10 cannot process are passed to a software controlled primary processing unit 11, including a central processing unit, CPU, 12, running software code which is more or less similar to the current software code run by the processors of most packet switches. If the packet looks like it is part of a new traffic stream, the central processing unit 12 provides the specialized hardware 10 with the necessary data parameters to deal with further packets from that packet traffic stream. Accordingly, any further packets seen from that data stream are dealt with automatically by the specialized hardware 10.

Detailed Description Text (11):

In operation, a packet switch normally examines the low level network header of an incoming packet at the input network, and removes that header from the packet. The packet is then passed to the software of the appropriate "protocol." The software generally checks the packet for errors, does certain bookkeeping on the packet, ensures that the packet is not violating flow or access controls, generates a route for the packet, and passes it to the output network. The output network constructs the outgoing network header, attaches it to the packet, and sends the packet on to

the next packet switch or other destination. At all stages in the process, the packet switch must guard against data congestion.

Detailed Description Text (12):

Most of these functions are identical on packets of the same stream and can therefore be separated from those functions which vary from packet to packet in the same packet stream. The repetitive functions can be performed once in software at CPU 12, at the time the hardware is first set up for a packet stream, that is, at the time the first packet of the stream is being processed. At this time, the hardware itself has very little that it is able to do. Thereafter, the hardware will handle all succeeding packets of the stream without any further intervention from the central processing unit.

Detailed Description Text (17):

In operation, a traffic stream is received and first identified by the CPU 12, as it receives the first packet of a new traffic stream from a CPU input buffer 26 connected to the input interconnect path 31. A free flow block 14 is selected to handle future packets of that traffic stream and all of the necessary information to handle the traffic stream, including the identification of the stream, is loaded into the pattern matching circuitry 16 and the control circuitry 18 of the selected flow block over the CPU bus 41.

Detailed Description Text (53):

Simultaneously, data from the input bus flows through a stripping circuit 254 which includes a counter and which discards the first "n" bytes of data (the header) allowing the remainder of the packet to pass through unmodified. The packet then passes to the control logic 18 where the higher level protocol functions such as check sum computation and hop count modification occur. The control logic 18, pattern matcher 16, and stripping circuit 254 have all been previously loaded with other necessary data from CPU 12 over bus 41. The input to the control device has a small amount of buffering to allow the control device to take more than one cycle when processing certain bytes in the data stream. The packet passing through this stage of processing may be modified; for example, this stage may abort further processing of the packet if an error is found, as described in more detail below. The packet then passes to a counter/truncate circuitry 260 which contains a counter loaded by the control logic over circuitry 262. The counter serves two functions: any unused trailer in the packet is discarded, and, if the packet is truncated, an error flag is raised over a line 264. The next stage of processing, a circuitry 266, prepends "n" bytes of data, the new output header, loaded from the CPU 12 in a similar manner to stripping circuit 254, to the packet as it passes therethrough. It also contains some buffering on the input to allow the new packet header to be inserted. In those instances where the new packet is substantially larger than the old one, the buffering is a necessity. The packet next passes to the output data buffer 20 which consists of a dual port (one read-only and one write-only) memory, along with a control logic 268 to keep track of the packets in the buffer. The buffer 20 is organized in a ring structure and a hardware queue of "t" buffer pointer/size pairs keeps track of the utilization of the buffer. Additional control circuitry within the buffer keeps track of the current start and end of the "free space". The packet then passes to an output multiplexor 274 which has output bus control logic and a set of drivers, one for each output bus in the output interconnect 52. When the flow block receives the "grant," for the appropriate output network interface 30, as described above, packets which are in the output buffer are read out and passed along the bus. Throughout the flow block, there are, in addition, data paths 276 which allow the CPU 12, over bus 41, to load memories, etc. in order to maintain proper operation of the flow block.

CLAIMS:

1. A high speed data packet switching circuit comprising:

a software controlled primary processing units,

a plurality of network interface units for receiving incoming data packet streams and for transmitting outgoing data packet streams, each of said data packet streams having a selected protocol and all of the data packets in a said stream having the identical protocol,

a plurality of data stream control circuits for concurrently receiving at least a portion of a header of the data packets and selectively processing the received packets only wherein each said data stream control circuit processes the data packets of one data stream having one of said selected protocol in response to previously generated electrical signals from the primary processing unit based upon header identification information in the at least first data packet of the new data packet stream for designating and initializing one of said data stream control circuits to process a remainder of the data packets of the new data packet stream,

means for interconnecting said primary processing unit, said plurality of interface units and said plurality of data stream control circuits,

said primary processing unit receiving from said network interface units, and for processing, at least a first one of the data packets of a new data packet stream and having means for generating said electrical signals means in each said designated and initialized data stream control circuit for receiving and processing only those data packets which include said header identification information upon which said designated and initializing is based.

8. The packet switching circuit of claim 1 further wherein said network interface unit comprises

a network interface circuit for communicating with a network channel in accordance with a said selected protocol and delivering data from the channel in a predetermined format, and

an interface adapter for receiving data from the channel through the network interface circuit in said predetermined format and for transmitting that data from the channel over the interconnecting means to said data stream control circuits and said primary processing unit, for receiving data, to be sent over a network channel, over said interconnecting means from the data stream control circuit and the primary processing unit and for delivering data received from said interconnecting means to said network interface circuit for transmission over a said network channel.

9. The packet switching circuit of claim 8 wherein said network interface unit further comprises

a single network special purpose hardware interface circuit having

means for communicating with a network channel,

means for transmitting received network data over the interconnecting means to said data stream control circuits and said primary processing unit,

means for receiving network data packets from the data stream control circuits and the primary processing unit, and

means for processing the received data packets for transmission over a network channel.

11. The packet switching circuit of claim 1 wherein said interconnecting means comprises

an input bus for interconnecting the outputs of said network interface units, the inputs of said data stream control circuits, and the primary processing unit, and

an output bus for interconnecting the outputs of said data stream control circuits, the inputs to said network interface units, and the primary processing unit.

12. The packet switching circuit of claim 11 wherein said interconnecting means further comprises a central processing unit bus interconnecting said data stream control circuits, said network interface units, and a central processing unit of said primary processing unit.

14. A high speed data packet switching method for switching data packet stream among communication paths comprising the steps of

receiving each packet stream from one of a plurality of networks,

processing at least a first packet of each received data packet stream using a software controlled, primary processing unit,

designating that performance of routine, repetitive header processing of the further packets of one of said received packet streams, said processing including packet forwarding processing to effect routing of said packet,

receiving and examining by each said high speed hardware circuitry at least a portion of each packet of each said received data packet stream, determining based on said examination of said at least a portion of each packet by each of said high speed hardware circuitry, which said high speed hardware circuitry has been designated to process each further packet of each received data packet stream, receiving in said designated high speed hardware circuitry said each further packet.

15. The high speed data packet switching method of claim 4 further comprising the step of

controlling at least the initialization of a said high speed hardware circuitry assigned to process a packet stream from the software controlled, primary processing unit.

16. A high speed data packet switching method comprising the steps of

receiving incoming packet streams from network interface units;

processing ones of the received data packets in response to a software controlled primary processing unit using a plurality of hardware data stream control circuits,

interconnecting the primary processing unit, the interface units, and the data stream control circuits for communications therebetween,

processing at least a first one of the data packets from the receiving step for each new data packet stream in the primary processing unit,

identifying, using the primary processing unit, one of the data stream control circuits for processing the incoming data packet stream,

determining by each said data stream control circuit the one data stream control circuit which will process each packet of that portion of said incoming data packet stream which is not processed by said primary processing unit,

processing that portion of a said data packet stream which is not processed by said primary processing unit by said identified data stream control circuit, and

outputting the results of the data stream control circuit processing and the primary processing unit processing to form an output data stream for transmission along a communications path.

17. A high speed data packet switching circuit for receiving data packet streams from a plurality of input communication paths and for transmitting data packet streams to a plurality of output communication paths, said circuit comprising

a plurality of network interface units for receiving the incoming data packet streams and for transmitting outgoing data packet streams,

a software controlled primary processing unit, having

a bus means,

a central processing unit,

a plurality of input storage units for receiving respectively each of said plurality of data streams from the network interface units and each input storage unit having its output connected to said bus means,

means for connecting the central processing unit to said bus means, and

a plurality of output storage units for receiving data from said central processing unit over said bus means, and for providing said data to said network interface units,

a plurality of data stream control circuits for manipulating data packet stream in response to the primary processing unit,

said data stream control circuits comprising

a pattern matching circuit responsive to pattern setting signals from the central processing unit and to incoming streams of data packets from said network interface units for identifying a data packet to be processed by said control circuit,

means for transferring identified data packets to said control circuit,

a processor responsive control circuit for controlling, in response to control signals sent by the primary processing unit, means for congestion control, and means for header stripping and prepending functions for the data stream control circuit, and

a data buffer responsive to said pattern matching circuit and the processor responsive control circuit for storing an incoming data packet stream from said control circuit and for outputting a stored data packet stream to be forwarded to a network interface unit,

means for interconnecting said primary processing unit, said plurality of network interface units and said plurality of data stream control circuits, and

said primary processing unit receiving from said network interface units at least a first one of the data packets of each new data packet stream and having means for designating those data packets of the stream which are not processing by the primary processing unit to be processed by a said data stream control circuit without further processing by said primary processing unit.

[Previous Doc](#)

[Next Doc](#)

[Go to Doc#](#)

WEST Search History

DATE: Wednesday, July 21, 2004

Hide?	<u>Set</u> <u>Name</u>	<u>Query</u>	<u>Hit</u> <u>Count</u>
	<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ</i>		
<input type="checkbox"/>	L15	L14 and buffer	35
<input type="checkbox"/>	L14	L13 and l6	44
<input type="checkbox"/>	L13	20010817	909
<input type="checkbox"/>	L12	L11 and (stream adj3 (ID or iderntifier or identification or map or mapping or table))	1312
<input type="checkbox"/>	L11	stream near5 processing	22311
<input type="checkbox"/>	L10	20010817	10
<input type="checkbox"/>	L9	L8 and (data adj2 stream)	10
<input type="checkbox"/>	L8	(2:1 or 3:1 or 3:2) adj2 multiplexing	94
<input type="checkbox"/>	L7	L6 and l2	3
<input type="checkbox"/>	L6	(packet or uniot or cell)adj3 (split or splitter or splitting or separator or separating or strip or stripping or parse or parsing or remove or removing)	35784
<input type="checkbox"/>	L5	l3 NOT l4	4
<input type="checkbox"/>	L4	L2 and (stream adj3 (ID or iderntifier or identification))	2
<input type="checkbox"/>	L3	L2 and (packet adj3 (process or processing))	4
<input type="checkbox"/>	L2	20010817	41
<input type="checkbox"/>	L1	(data adj3 stream) near8 (multiplex or multiplexing) near8 bandwidth	66

END OF SEARCH HISTORY

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)
[First Hit](#) [Fwd Refs](#)

☐ [Generate Collection](#)

L23: Entry 1 of 1

File: USPT

Aug 13, 1996

DOCUMENT-IDENTIFIER: US 5546390 A

TITLE: Method and apparatus for radix decision packet processing

Application Filing Date (1):
19941229

Brief Summary Text (23):

The use of multiple processors to avoid the "processor bottleneck" provides some gains but again has limits. Given a code path to forward a data packet, it is not plausible to split that path into more than a few stages. Typically these stages would involve network input, protocol functions, and network output. The basis for this limitation is the overhead incurred to interface the different processors beyond a limited number of task divisions. That is, after a certain point, the increase in interface overhead outweighs the savings obtained from the additional stage. This is particularly true because of the need to tightly integrate the various components; for example, congestion control at the protocol level requires close coordination with the output device. Also, the interface overhead costs are made more severe by the complication of the interface which is required.

CLAIMS:

1. A method of processing a protocol data unit in a communication network, comprising the device-implemented steps of:

(a) optimizing a decision process by selectively examining only those significant bits of a protocol data unit received from the communication network which affect the decision process, these decision-significant bits comprising at least two non-contiguous bits of the protocol data unit;

(b) validating the decision process by comparing a portion of the received protocol data unit with a predetermined tuple, the predetermined tuple consisting of known values for a specific portion of the protocol data unit including the at least two non-contiguous decision-significant bits; and

(c) generating at least one associated directive for the protocol data unit based upon the validated decision process, the at least one associated directive specifying subsequent processing requirements of the protocol data unit.

7. A method of radix-type decision processing a protocol data unit in a communication network, comprising the device-implemented steps of:

(a) selectively examining only those significant bits of a protocol data unit received from the communication network which affect a radix tree-type decision process, these decision-significant bits comprising at least two non-contiguous bits of the protocol data unit;

(b) grouping the decision-significant bits together into decision groups; and

(c) making decisions in the decision process based on decision groups rather than

individual decision-significant bits.

10. A protocol data unit preprocessing device for use in a communication network to optimize a decision process related to protocol data units within the communication network, the preprocessing device comprising:

(a) filter means for selectively examining only those bits of a protocol data unit received from the communication network which affect the decision process, these decision-significant bits comprising at least two non-contiguous bits of the protocol data unit;

(b) validation means, operatively coupled to the filter means, for validating the decision process by comparing a portion of the protocol data unit with a predetermined tuple, the predetermined tuple consisting of known values for a specific portion of the protocol data unit including the at least two non-contiguous decision-significant bits; and

(c) generation means, operatively coupled to the validation means, for generating at least one associated directive for the protocol data unit based upon the validated decision process, the at least one associated directive specifying subsequent processing requirements of the protocol data unit.

22. A radix-based decision processor for use in a communication network, comprising:

(a) filter means for selectively examining only those significant bits of a protocol data unit received from the communication network which affect a radix-type decision process, these decision-significant bits comprising at least two non-contiguous bits of the protocol data unit;

(b) grouping means, operatively coupled to the filter means, for grouping the decision-significant bits together into decision groups; and

(c) decision means, operatively coupled to the grouping means, for making decisions in the decision process based on decision groups rather than individual decision-significant bits.

[Previous Doc](#)

[Next Doc](#)

[Go to Doc#](#)

[Previous Doc](#)[Next Doc](#)
[First Hit](#)[Go to Doc#](#)

Generate Collection

L30: Entry 2 of 2

File: DWPI

Jun 3, 2003

DERWENT-ACC-NO: 1997-435451

DERWENT-WEEK: 200343

COPYRIGHT 2004 DERWENT INFORMATION LTD

TITLE: Network data communication apparatus for computer network - implements packet segmentation and reassembly for cell based switching on backplane cell bus, which has several packet processing units coupled to it, where each unit hosts several LAN ports

Basic Abstract Text (1):

The apparatus includes a first data port which is coupled to receive an incoming network data packet. A data packet segmentation unit is in communication with the first port for segmenting the incoming data packet into several fixed size data cells. A cell bus is in communication with the data packet segmentation unit for conveying the fixed size data cells.

Basic Abstract Text (2):

The data packet reassembly unit is in communication with the cell bus for receiving the data cells and for reassembling the data cells into a network data packet. A second port is in communication with the data packet reassembly unit for transmitting the network data packet from the data communication apparatus.

PF Application Date (1):19961108PF Application Date (2):19961108PF Application Date (5):19961108PF Application Date (6):19961108PF Application Date (8):19960223PF Application Date (9):19961108PF Application Date (10):19961108Equivalent Abstract Text (1):

The apparatus includes a first data port which is coupled to receive an incoming network data packet. A data packet segmentation unit is in communication with the first port for segmenting the incoming data packet into several fixed size data cells. A cell bus is in communication with the data packet segmentation unit for conveying the fixed size data cells.

Equivalent Abstract Text (2):

The data packet reassembly unit is in communication with the cell bus for receiving the data cells and for reassembling the data cells into a network data packet. A second port is in communication with the data packet reassembly unit for transmitting the network data packet from the data communication apparatus.

[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)[First Hit](#)

Generate Collection

L32: Entry 1 of 1

File: PGPB

May 16, 2002

DOCUMENT-IDENTIFIER: US 20020057669 A1

TITLE: Multi-layered packet processing deviceApplication Filing Date:

20010706

Detail Description Paragraph:

[0025] When the packet is transferred while being encapsulated in the ATM AAL5, as shown in the Table 1, the structure shown in FIG. 2 handles the packet at a hardware level. FIG. 2 shows the structure of a main module interface 10, an ATM reassembly 20, an ingress IP processor 22, a GTP (GPRS Tunnel Protocol) and UDP (User Datagram Protocol) processor 40, a lookup processor 50, an egress IP processor 24, a segmentation 60, and an ATM switch interface 70, all of which are connected in a pipelined structure.

Detail Description Paragraph:

[0030] Upon receipt of either the first packet (P1) updated with the destination address, or the third packet (P3) having the valid index value written in its tag, the egress IP processor 24 requests a remainder of a payload which is temporarily stored in the ATM reassembly 20, and outputs the remainder of the payload to the segmentation 60, after either the first packet (P1) updated with the destination address is output, or after the third packet (P3) having the valid index value written in its tag is output.

Detail Description Paragraph:

[0031] When the remainder of the payload is received from the egress IP processor 24, together with the first packet (P1) updated with the destination address, the segmentation 60 looks up an appropriate VPI/VCI value, segments the first packet (P1) into the ATM cells, and outputs the ATM cells to the main module interface 10. When the remainder of the payload is received together with the third packet (P3) having the valid index value written in its tag, the segmentation 60 segments the third packet (P3) into the ATM cells, and outputs the ATM cells with the tag to the ATM switch interface 70.

Detail Description Paragraph:

[0032] Here, the ATM reassembly 20 has two memory blocks, i.e., a connection memory 20-2 and a cell memory 20-1, as shown in FIG. 3. The cell memory 20-1 stores a remainder of the packet that is separated from the ATM reassembly 20. The GTP and UDP processor 40 has a TEID lookup table 40-1. The lookup processor 50 has an IP address lookup table 50-1. The segmentation 60 has an index table 60-1. The whole structure of the packet processing device is shown in FIG. 3.

Detail Description Paragraph:

[0039] Referring back to the FIGS. 4B and 6, the operation of the IP processor 30 will be described. When the egress IP processor 24 receives the third packet (P3) from the lookup processor 50 (step S4), the egress IP processor 24 checks whether the valid index value of the tag is set or not. When the valid index value of the tag is not set, the egress IP processor 24 requests from the ATM reassembly 20, the remainder of the third packet (P3), by using the packet address field (step

S5). When it is determined that the remainder of the third packet (P3) has been received (step S6), the third packet (P3) is transferred to the segmentation 60, and the remainder of the third packet (P3), which is requested by the egress IP processor 24, is successively output to the segmentation 60, upon arrival to the egress IP processor 24 (step S7). The other Start Of Packets (SOP) are not transferred to the segmentation 60 until the remainder of the third packet (P3) is completely transferred to the segmentation 60.

Detail Description Paragraph:

[0040] When the segmentation 60 receives the third packet (P3) having a valid index value written in its tag together with the remainder of the payload, the segmentation 60 outputs the tag-included ATM cells to the ATM switch interface 70. When the segmentation 60 receives the first packet (P1) updated with the destination address and the remainder of the payload, the segmentation 60 looks up an appropriate VPI/VCI value, segments the first packet (P1) into the ATM cells, and outputs the ATM cells to the interface 10.

CLAIMS:

2. The multi-layered packet processing device of claim 1, wherein the plurality of packet processing portions comprise: a packet separating processor for outputting a packet to be analyzed, by sequentially including a tag in part of the data packet transferred from the interface, the packet separating processor for storing a remainder of the packet to be analyzed, which is left after the packet to be analyzed is output; a plurality of header analyzing processors for sequentially analyzing the packet to be analyzed transferred from the packet separating processor, according to a header encapsulated in the packet to be analyzed, and then reflecting an analyzed result in the tag of the packet to be analyzed, and outputting an analyzed packet; a packet reassembling processor for requesting the remainder of the packet to be analyzed stored in the packet separating processor, when the packet reassembling processor receives the analyzed packet output from the plurality of header analyzing processors, and outputting the analyzed packet together with the requested remainder of the packet to be analyzed, as a complete data packet; and an output processor for determining an output route of the complete data packet by analyzing output route information reflected in the tag of the complete data packet transferred from the packet reassembling processor, and outputting the complete data packet according to the determined output route.

4. The multi-layered packet processing device of claim 3, wherein the output processor segments the complete data packet transferred from the packet reassembling processor into the ATM cells, and outputs the ATM cells.

5. The multi-layered packet processing device of claim 2, wherein the plurality of header analyzing processors comprise: an internet protocol (IP) header analyzing processor for determining whether a destination address of the packet to be analyzed matches a system address, and outputting an IP header-removed first packet when the destination address of the packet to be analyzed matches the system address; a protocol transmission type header analyzing processor for analyzing a protocol transmission type header of the IP header-removed first packet, reflecting the analyzed result in the tag of the packet to be analyzed, and outputting a second packet from which the protocol transmission type header is removed; and a lookup processor for updating the destination address of the packet to be analyzed, which is transferred from the protocol transmission type header analyzing processor, outputting an updated packet to be analyzed to the packet reassembling processor, and outputting the second packet, which is transferred from the protocol transmission type header analyzing processor, together with a bypass signal, without processing.

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)

THIS PAGE BLANK (USPTO)

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)
[First Hit](#) [Fwd Refs](#)



Generate Collection

L34: Entry 1 of 1

File: USPT

Jun 10, 1997

DOCUMENT-IDENTIFIER: US 5638367 A

**** See image for Certificate of Correction ****

TITLE: Apparatus and method for data packing through addition

Detailed Description Text (25):

Referring now to FIGS. 7a-7i, in order to clarify the operations of the present invention, a specific example has been created for the sole purpose of explaining the byte packing with addition technique. This specific example should not be construed in any way as a limitation on the scope of the present invention.

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)

[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)[First Hit](#)

L14: Entry 12 of 44

File: PGPB

Jan 10, 2002

DOCUMENT-IDENTIFIER: US 20020003795 A1

TITLE: IN-LINE PACKET PROCESSING

Application Filing Date:

19980804

Summary of Invention Paragraph:

[0011] The step of receiving the data packet includes receiving a plurality of data packets for processing from a plurality of input ports representing a plurality of streams of data to be routed through the router. The step of pre-processing the data packet includes dividing the data packet into fixed length cells and parsing the L2 header associated with the first cell of the data packet prior to receipt of the entire data packet. The step of parsing the L2 header includes examining the L2 header for errors, and identifying the start of a next layer header in the data packet.

Summary of Invention Paragraph:

[0013] The method can include snooping while the cells are being written to the queue and parsing the L3 header including examining the L3 header for errors. A data packet can be dropped if errors are detected in the L2 header during L2 header parsing without storing a cell associated with the data packet in the queue.

Summary of Invention Paragraph:

[0018] The router can further include a cell packetizer operable to divide the data packet into fixed length cells and a L2 parsing engine operable to examine the L2 header associated with the first cell of the data packet prior to receipt of the entire data packet. The L2 parsing engine is operable to examine the L2 header for errors and identify the start of a next layer header in the data packet.

Summary of Invention Paragraph:

[0020] The router can further include a L3 parsing engine operable to snoop while the cells are being written to the queue and parse the L3 header including examining the L3 header for errors. The L2 parser engine is operable to drop a data packet if errors are detected in the L2 header during L2 header parsing without storing a cell associated with the data packet in the queue.

Detail Description Paragraph:

[0062] Associated with the L2 header parser is a per stream L2 state queue 419. Per stream state queue 499 stores flags associated with a micro-code starting address, priority (precedence) flag for the stream, an interface index mapping to one or more logical interfaces, virtual connection stream state information and channel stream state information. The per stream state queue stores information associated with each stream so as to assure continuity in steam processing.

Detail Description Paragraph:

[0078] The flags store information required for the efficient down-stream processing of a given packet. In one implementation, the L2 flags derived during L2 processing include a packet loss priority flag, a send packet to processor flag, a sample packet flag and a physical multicast flag. The L3 flags derived during L3

header processing include an option flag, packet priority flag, transmission control protocol (TCP) flag, protocol type flag, and DF (don't fragment) flag.

CLAIMS:

14. The method of claim 1 where the step of pre-processing the data packet includes dividing the data packet into fixed length cells and parsing the L2 header associated with the first cell of the data packet prior to receipt of the entire data packet.

17. The method of claim 16 including snooping while the cells are being written to the queue and parsing the L3 header including examining the L3 header for errors.

33. The router of claim 19 further including a cell packetizer operable to divide the data packet into fixed length cells and a L2 parsing engine operable to examine the L2 header associated with the first cell of the data packet prior to receipt of the entire data packet.

36. The router of claim 35 further including a L3 parsing engine operable to snoop while the cells are being written to the queue and parse the L3 header including examining the L3 header for errors.

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)

[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)[First Hit](#)

Generate Collection

L35: Entry 1 of 2

File: PGPB

Jan 10, 2002

DOCUMENT-IDENTIFIER: US 20020003795 A1

TITLE: IN-LINE PACKET PROCESSINGApplication Filing Date:

19980804

Detail Description Paragraph:

[0049] Referring to FIG. 3d, the packing process is performed in two phases. In a first phase, 8 bit or 32 bit data is accumulated in a bit packing queue 381a. Bit packing queue 381a includes stream bit packing queues (381a-1 thru 381a-16), one for each stream. In one implementation, bit packing queue 381a includes 16 stream bit packing queues (when supporting 16 OC-3 streams). Bit packing queue 381a includes a controller 385a for streaming data words from the bit packing queue to a byte packing queue 381b. Each stream bit packing queue can be sized to hold two or more words of data, and in one implementation are sized to hold three 64 bit words of data (8 byte data words). Associated with each data word stored in the stream bit packing queues are control flags which are received along with the stream data. In one embodiment, five (5) bits of flag data are stored with each 64 bit data word in each stream bit packing queue. The five flag data bits are passed with each 64 bit word from the bit packing queue 381a to the byte packing queue 381b.

Detail Description Paragraph:

[0050] In one embodiment the control flags include an end of packet without error flag (1 bit), a end of packet with error flag (1 bit), and a last byte pointer (3 bits). The size of the last byte pointer indicates the last byte in the eight (8) byte word of data that contains data in the transfers between the bit packing queue 381a and byte packing queue 381b. In one implementation, line input interface 300 screens the incoming packets and generates the control flag data. The control flag data can be advantageously used in screening packets prior to transfer to the global memory. More specifically, errors detected by the line input interface 300 are reconciled during L2 header processing. A packet that is written to segmentation buffer 387 that has an error flag set is never transferred by the packetizer 391 to payload buffer 388. Data associated with the packet is overwritten in time by a next 64 byte data word from byte packing queue 381b. Other early error detection methods are described in greater detail below in association with L2 and L3 processing.

Detail Description Paragraph:

[0051] Based on the availability of data in the individual stream bit packing queues, a single 64 bit word is transferred per clock cycle by controller 385a from bit packing queue 381a to byte packing queue 381b. Controller 385a cycles through the individual bit packing queues in a round-robin fashion to transfer available data words to byte packing queue 381b.

Detail Description Paragraph:

[0052] In a second phase of the packing process, byte packing queue 381b accumulates eight (8) byte portions of data (64 bit data words) prior to segmentation. Byte packing queue 381b includes stream byte packing queues (381b-0 thru 381b-15), one for each stream. In one implementation, byte packing queue 381b

includes 16 stream byte packing queues to support 16 streams of OC-3 data. Depending on the format of the stream data received, a lesser number of the stream byte packing queues can be used. Byte packing queue 381b is sized to support the input bandwidth, which in one implementation is 2.4 Gbps. Byte packing queue 381b is configurable to support a variety of input stream configurations. Byte packing queue 381b can be a flexible buffer whose resources are dynamically allocated at start-up depending on the input configuration.

Detail Description Paragraph:

[0053] Byte packing queue 381b includes a cell dispatcher 385b for streaming data words from byte packing queue 381b to segmentation buffer 387. Each stream byte packing queue can be sized to hold N or more words of data, and in one implementation each is sized to hold eighteen (18) eight (8) byte data portions (64 bit data words). Associated with each data word stored in the stream byte packing queues are control flags which are received along with the stream data.

Detail Description Paragraph:

[0054] Depending on the input configuration of the stream data received, a lesser number of divisions for byte packing queue 381b may be required. For example, when supporting four OC-12 streams, byte packing queue 381b is configured with four byte packing queues to accumulate 64 bit data words for transfer to segmentation buffer 387.

Detail Description Paragraph:

[0055] Cell dispatcher 385b operates in a round-robin fashion to cycle through the various individual stream byte packing queues to extract data words for transfer to segmentation buffer 387. In one embodiment, eight (8) byte read cycles are used to transfer data from the individual byte packing queues to segmentation buffer 387.

Detail Description Paragraph:

[0056] Each byte packing queue signals to the cell dispatcher when an appropriate number of data words have been accumulated and thus the respective byte packing queue is ready for servicing. A stream byte packing queue initiates a signal to cell dispatcher 385b indicating data is ripe for transfer upon the occurrence of one of three trigger conditions: upon receipt of the first 96 bytes of a new packet; upon receipt of an end of packet flag; or upon receipt of 64 bytes of data which do not comprise the beginning or the end of a packet. The flags received along with the data words from the bit packing queue 381a are used to evaluate the data words received from byte packing queue 381a to determine when the various conditions have been satisfied.

Detail Description Paragraph:

[0057] The individual stream byte packing queues are sized to ensure that entries received from bit packing queue 381a are not lost while waiting for service by cell dispatcher 385b. In an embodiment that includes 16 OC-3 input streams, all of the stream byte packing queues are identically sized to include eighteen 64 bit words.

Detail Description Paragraph:

[0059] As described above, data words are transferred from byte packing queue 381b to segmentation buffer 387 as they become available. In addition, coincident with the transfer, the first 32 bytes of data packet (four 8 byte data words) are also transferred to L2 header buffer 383 of L2 pattern match decoder 382. For middle packets which include up to 64 bytes of data in the middle of a packet, cell dispatcher 385b disables the transfer of data to the L2 header buffer and no L2 header processing is required. The present invention processes packets in-line. In-line processing includes two components, one at the input prior to storage of a packet (or portion of the packet in packet memory, i.e. memory 104 of FIG. 2b) and the other at the output after packets are read from packet memory. The input component conditionally strips the L2 header, finds the start of the L3 header and checks the L3 header for errors. In addition, other early error verifications can

be performed to assure that bad packets are dropped prior to storage in packet memory. The output component attaches a new L2 header to a packet and updates certain L3 header fields as required. In-line processing significantly increases throughput in the router allowing for the early dumping of packets prior to storage in packet memory. The input component is described immediately below. The output component is described later in the specification.

Detail Description Paragraph:

[0061] In addition, the L2 header parser examines portions of the L2 header for errors. Errors arising from unrecognized L2 headers, unconfigured L2 connections, or other L2 errors are immediately identified and dropped prior to being written to memory. In one implementation packets are dropped by never transferring the packet out of the segmentation buffer 387. The dropped packet is overwritten in time by a next 64 byte data word from byte packing queue 381b.

Detail Description Paragraph:

[0063] Segmentation buffer 387 is sized to accommodate up to eight data word (64 bytes) in one implementation. Associated with the segmentation buffer is a selectable start read pointer that indicates the location in the segmentation buffer to begin read operations (when reading data from the segmentation engine by packetizer 391). Data bytes are read from byte packing queue 381b and stored in segmentation buffer 387. Subsequent byte transfers (beyond the first 64 bytes) from byte packing queue 381b can result in a wrap around condition. Wrap around results in the overwriting of a portion of the contents of the segmentation buffer. Wrap around may arise when a portion of the L2 header is overwritten in accordance with the strip offset determined as part of L2 header processing. A circular buffer can be used to minimize the size of the buffer required to support the line rate processing.

Detail Description Paragraph:

[0064] The segmentation buffer provides temporary storage of the data words transferred from the byte packing queue while the L2 processing completes. Transfer from segmentation buffer 387 by cell packetizer 391 to cell payload queue 388 is initiated upon completion of the L2 processing and delivery of the offset information to cell packetizer 391. Cell dispatcher 385b triggers the transfer of middle data words in a packet (all data words after the first data word associated with a packet).

Detail Description Paragraph:

[0068] As described above, the offset information is derived from the L2 packet header. The offset information can be provided in the form of a pointer pointing to a particular location in the data stored in segmentation buffer 387. The pointer can be used to indicate the particular byte in the data transferred from the segmentation buffer that marks the beginning the next layer header. Cell packetizer 391 can discard those portions of the data that belong to the L2 header necessitating further reads from the byte packing queue 381b to fill a data portion of a cell (64 byte) in cell payload queue 388.

Detail Description Paragraph:

[0069] The offset may not arise exactly on an eight byte boundary necessitating the storage of overflow data. To facilitate block transfers from byte packing queue 381b to segmentation buffer 387 (8 byte blocks), an overflow queue is provided. Segmentation state queue 401 includes a queue sized to contain N-1 bytes of data for each stream, where N is equal to the size of the block transfers from byte packing queue 381b to segmentation buffer 387. Extra bytes that are required to be read from byte packing queue 381b to facilitate the filling of a cell in cell payload queue 388 are stored in segmentation state queue 401.

Detail Description Paragraph:

[0071] The position in cell payload queue 388 to which the cell is written is

controlled by buffer pool manager 393. Cell dispatcher 385b receives feedback from buffer pool manager 393 as entries are extracted from cell payload queue 388. A slot must be available in cell payload queue 388 prior to the packetizing of the cell data and extraction from byte packing queue 381b. Buffer pool manager includes a pointer that indicates the next available cell in the cell payload queue that can be written to by cell packetizer 391. As each cell is written into cell payload queue 388 an associated header is written into the cell header queue 390.

[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)[First Hit](#)

Generate Collection

L14: Entry 1 of 44

File: PGPB

Dec 26, 2002

DOCUMENT-IDENTIFIER: US 20020196374 A1

TITLE: DIGITAL MULTI-MEDIA DEVICE AND METHOD RELATING THERETO

*Stripping unwanted
packets from a stream*

Abstract Paragraph:

A method of transmitting a transport stream over an IEEE 1394 Serial Bus comprising stripping out of the transport stream identifiable unrequired data so as to reduce the bandwidth of the transport stream, a digital multi-media receiver for transmitting a transport stream to a Conditional Access Module on an IEEE 1394 Serial Bus including a reader for reading the contents of the transport stream and identifying data which is not required for processing by a Conditional Access Module, a stripper for stripping out at least some of the identified unrequired data from the transport stream and a transmitter for transmitting all of the remaining data of the transport stream which has not been stripped and a network of digital multi-media devices connected by means of an IEEE 1394 Serial Bus whereby a first device sends such a stripped transport stream to a second device over the IEEE 1394 Serial Bus.

Application Filing Date:

19990422

Summary of Invention Paragraph:

[0021] a reader for reading the contents of a transport stream and identifying data which is not required for processing by a Conditional Access Module;

Summary of Invention Paragraph:

[0027] the first device includes a reader for reading the contents of a transport stream and identifying data which is not required for processing by the second device;

Brief Description of Drawings Paragraph:

[0121] FIGS. 11(a) and (b) illustrate a transport stream and associated table.

Detail Description Paragraph:

[0146] The IEEE 1394 specification defines the lower layers for the carriage of isochronous data, these being the physical and link layers as described above. The IEC 1883 protocols provide mechanisms to allow the efficient transport of AV data utilising a Common Isochronous Packet (CIP) header. This allows AV data packets to be split up for transport over the IEEE 1394 bus and also has fields to signal data format (standard or high definition video data, 50 or 60 Hz field rate). The IEC 1883 specification also provides the concept of logical channels and plugs for the carriage and connection of AV data between devices on the IEEE 1394 bus.

Detail Description Paragraph:

[0217] The use of Conditional Access Modules in conjunction with the IEEE Serial Bus brings about the possibility of communication of another type not previously considered for the IEEE 1394 Serial Bus. As an example, rather than return a descrambled transport stream to the host receiver 30, the transport stream can be passed on to a second Conditional Access Module (not shown) for further processing or descrambling of other virtual channels of the transport stream. To do this, it

is proposed that the host receiver 30 contacts the isochronous resource manager to request a third isochronous channel, configures the output control register of the first module 38 to remove the return connection between the first module 38 and the host 30 and sets up a new connection between the first 38 and second modules using the second isochronous channel. The host 30 then configures the input plug register of the second module to receive a transport stream on the second isochronous channel and configures the output plug control register of the second module to set up a connection between the second module and the host 30 using the third isochronous channel. It then configures its own input plug control register to receive the transport stream on the third isochronous channel.

Detail Description Paragraph:

[0226] A transport stream usually contains a table or map, for example MPEG Program Specific Information (PSI), indicating the various program streams present and other identifying maps. However, since not all the data in the transport stream need be identified in the table, a host receiver may not be able to identify the presence of certain data.

Detail Description Paragraph:

[0228] FIG. 11(a) illustrates schematically time multiplexed data streams, together with an associated table.

Detail Description Paragraph:

[0230] As will be apparent, from the table, a device can determine the content of associated data streams, for example that data stream 1 relates to program Channel D. However, as illustrated, not all the data streams need be identified in the table.

CLAIMS:

1. A digital multi-media receiver comprising: an output for an IEEE 1394 Serial Bus for transmitting a transport stream to a Conditional Access Module on the bus; a reader for reading the contents of a transport stream and identifying data which is not required for processing by a Conditional Access Module; a stripper for stripping out at least some of the identified unrequired data from the transport stream; and a transmitter for transmitting through the output all of the remaining data of the transport stream which has not been stripped.

6. A network of digital multi-media devices connected by means of an IEEE 1394 Serial Bus, the network comprising: a device for transmitting a transport stream over the IEEE 1394 Serial Bus; and a second device for receiving a transport stream over the IEEE 1394 Serial Bus; wherein the first device includes a reader for reading the contents of a transport stream and identifying data which is not required for processing by the second device; a stripper for stripping out at least some of the identified unrequired data from the transport stream; and a transmitter for transmitting to the second device over the IEEE 1394 Serial Bus all of the remaining data of the transport stream which has not been stripped.

[Previous Doc](#)

[Next Doc](#)

[Go to Doc#](#)

Hit List

Clear

Generate Collection

Print

Fwd Refs

Bkwd Refs

Generate OACS

Search Results - Record(s) 1 through 10 of 18 returned.☐ 1. Document ID: US 20020003795 A1

L14: Entry 1 of 18

File: PGPB

Jan 10, 2002

PGPUB-DOCUMENT-NUMBER: 20020003795

PGPUB-FILING-TYPE: new

DOCUMENT-IDENTIFIER: US 20020003795 A1

TITLE: IN-LINE PACKET PROCESSING

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWC	Draw D
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	-----	--------

☐ 2. Document ID: US 20010049744 A1

L14: Entry 2 of 18

File: PGPB

Dec 6, 2001

PGPUB-DOCUMENT-NUMBER: 20010049744

PGPUB-FILING-TYPE: new

DOCUMENT-IDENTIFIER: US 20010049744 A1

TITLE: High-speed data processing using internal processor memory space

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWC	Draw D
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	-----	--------

☐ 3. Document ID: US 20010027496 A1

L14: Entry 3 of 18

File: PGPB

Oct 4, 2001

PGPUB-DOCUMENT-NUMBER: 20010027496

PGPUB-FILING-TYPE: new

DOCUMENT-IDENTIFIER: US 20010027496 A1

TITLE: Passing a communication control block to a local device such that a message is processed on the device

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWC	Draw D
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	-----	--------

☐ 4. Document ID: US 20010023460 A1

L14: Entry 4 of 18

File: PGPB

Sep 20, 2001

PGPUB-DOCUMENT-NUMBER: 20010023460
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20010023460 A1

TITLE: Passing a communication control block from host to a local device such that a message is processed on the device

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw D
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	--------

☐ 5. Document ID: US 6650640 B1

L14: Entry 5 of 18

File: USPT

Nov 18, 2003

US-PAT-NO: 6650640
DOCUMENT-IDENTIFIER: US 6650640 B1

TITLE: Method and apparatus for managing a network flow in a high performance network interface

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw D
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	--------

☐ 6. Document ID: US 6606301 B1

L14: Entry 6 of 18

File: USPT

Aug 12, 2003

US-PAT-NO: 6606301
DOCUMENT-IDENTIFIER: US 6606301 B1

**** See image for Certificate of Correction ****

TITLE: Method and apparatus for early random discard of packets

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw D
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	--------

☐ 7. Document ID: US 6483804 B1

L14: Entry 7 of 18

File: USPT

Nov 19, 2002

US-PAT-NO: 6483804
DOCUMENT-IDENTIFIER: US 6483804 B1

TITLE: Method and apparatus for dynamic packet batching with a high performance network interface

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw D
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	--------

☐ 8. Document ID: US 6480489 B1

L14: Entry 8 of 18

File: USPT

Nov 12, 2002

US-PAT-NO: 6480489

DOCUMENT-IDENTIFIER: US 6480489 B1

TITLE: Method and apparatus for data re-assembly with a high performance network interface

Full	Title	Citation	Front	Review	Classification	Date	Reference	Abstract	Claims	Keywords	Drawings
------	-------	----------	-------	--------	----------------	------	-----------	----------	--------	----------	----------

☐ 9. Document ID: US 6453360 B1

L14: Entry 9 of 18

File: USPT

Sep 17, 2002

US-PAT-NO: 6453360

DOCUMENT-IDENTIFIER: US 6453360 B1

TITLE: High performance network interface

Full	Title	Citation	Front	Review	Classification	Date	Reference	Abstract	Claims	Keywords	Drawings
------	-------	----------	-------	--------	----------------	------	-----------	----------	--------	----------	----------

☐ 10. Document ID: US 6434620 B1

L14: Entry 10 of 18

File: USPT

Aug 13, 2002

US-PAT-NO: 6434620

DOCUMENT-IDENTIFIER: US 6434620 B1

TITLE: TCP/IP offload network interface device

Full	Title	Citation	Front	Review	Classification	Date	Reference	Abstract	Claims	Keywords	Drawings
------	-------	----------	-------	--------	----------------	------	-----------	----------	--------	----------	----------

Clear

Generate Collection

Print

Fwd Refs

Bkwd Refs

Generate OACS

Term	Documents
PACKET	158072
PACKETS	98523
PROCESSING	2360659
PROCESSINGS	24582
REASSEMBLY	15188
REASSEMBLIES	79
REASSEMBLYS	0
REASSEMBLING	6672
REASSEMBLINGS	3
(12 AND (PACKET ADJ3 (REASSEMBLING OR REASSEMBLY OR PROCESSING))) .PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD.	18
(L12 AND (PACKET ADJ3 (PROCESSING OR REASSEMBLY OR	

Hit List

Clear

Generate Collection

Print

Fwd Refs

Bkwd Refs

Generate OACS

Search Results - Record(s) 11 through 18 of 18 returned.

☐ 11. Document ID: US 6393487 B2

L14: Entry 11 of 18

File: USPT

May 21, 2002

US-PAT-NO: 6393487

DOCUMENT-IDENTIFIER: US 6393487 B2

TITLE: Passing a communication control block to a local device such that a message is processed on the device

Full	Title	Citation	Front	Review	Classification	Date	Reference	Abstract	Claims	KWIC	Draw D
------	-------	----------	-------	--------	----------------	------	-----------	----------	--------	------	--------

☐ 12. Document ID: US 6389468 B1

L14: Entry 12 of 18

File: USPT

May 14, 2002

US-PAT-NO: 6389468

DOCUMENT-IDENTIFIER: US 6389468 B1

TITLE: Method and apparatus for distributing network traffic processing on a multiprocessor computer

Full	Title	Citation	Front	Review	Classification	Date	Reference	Abstract	Claims	KWIC	Draw D
------	-------	----------	-------	--------	----------------	------	-----------	----------	--------	------	--------

☐ 13. Document ID: US 6356951 B1

L14: Entry 13 of 18

File: USPT

Mar 12, 2002

US-PAT-NO: 6356951

DOCUMENT-IDENTIFIER: US 6356951 B1

**** See image for Certificate of Correction ****

TITLE: System for parsing a packet for conformity with a predetermined protocol using mask and comparison values included in a parsing instruction

Full	Title	Citation	Front	Review	Classification	Date	Reference	Abstract	Claims	KWIC	Draw D
------	-------	----------	-------	--------	----------------	------	-----------	----------	--------	------	--------

☐ 14. Document ID: US 6347347 B1

L14: Entry 14 of 18

File: USPT

Feb 12, 2002

US-PAT-NO: 6347347

DOCUMENT-IDENTIFIER: US 6347347 B1

TITLE: Multicast direct memory access storing selected ones of data segments into a first-in-first-out buffer and a memory simultaneously when enabled by a processor

Full	Title	Citation	Front	Review	Classification	Date	Reference	Abstract	Claims	KWIC	Drawings
------	-------	----------	-------	--------	----------------	------	-----------	----------	--------	------	----------

☐ 15. Document ID: US 6334153 B1

L14: Entry 15 of 18

File: USPT

Dec 25, 2001

US-PAT-NO: 6334153

DOCUMENT-IDENTIFIER: US 6334153 B1

TITLE: Passing a communication control block from host to a local device such that a message is processed on the device

Full	Title	Citation	Front	Review	Classification	Date	Reference	Abstract	Claims	KWIC	Drawings
------	-------	----------	-------	--------	----------------	------	-----------	----------	--------	------	----------

☐ 16. Document ID: US 6247060 B1

L14: Entry 16 of 18

File: USPT

Jun 12, 2001

US-PAT-NO: 6247060

DOCUMENT-IDENTIFIER: US 6247060 B1

TITLE: Passing a communication control block from host to a local device such that a message is processed on the device

Full	Title	Citation	Front	Review	Classification	Date	Reference	Abstract	Claims	KWIC	Drawings
------	-------	----------	-------	--------	----------------	------	-----------	----------	--------	------	----------

☐ 17. Document ID: US 6154460 A

L14: Entry 17 of 18

File: USPT

Nov 28, 2000

US-PAT-NO: 6154460

DOCUMENT-IDENTIFIER: US 6154460 A

TITLE: Data packet transmission system and method

Full	Title	Citation	Front	Review	Classification	Date	Reference	Abstract	Claims	KWIC	Drawings
------	-------	----------	-------	--------	----------------	------	-----------	----------	--------	------	----------

☐ 18. Document ID: US 6078733 A

L14: Entry 18 of 18

File: USPT

Jun 20, 2000

US-PAT-NO: 6078733

DOCUMENT-IDENTIFIER: US 6078733 A

TITLE: Network interface having support for message processing and an interface to a message coprocessor

Full	Title	Citation	Front	Review	Classification	Date	Reference	Statistics	Abstracts	Claims	KWIC	Draws
------	-------	----------	-------	--------	----------------	------	-----------	------------	-----------	--------	------	-------

Clear	Generate Collection	Print	Fwd Refs	Bkwd Refs	Generate OACS
-------	---------------------	-------	----------	-----------	---------------

Term	Documents
PACKET	158072
PACKETS	98523
PROCESSING	2360659
PROCESSINGS	24582
REASSEMBLY	15188
REASSEMBLIES	79
REASSEMBLYS	0
REASSEMBLING	6672
REASSEMBLINGS	3
(12 AND (PACKET ADJ3 (REASSEMBLING OR REASSEMBLY OR PROCESSING))) .PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD.	18
(L12 AND (PACKET ADJ3 (PROCESSING OR REASSEMBLY OR REASSEMBLING))) .PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD.	18

Display Format:

[Previous Page](#)

[Next Page](#)

[Go to Doc#](#)